What is claimed is:

1. A data processing method for application layer based on a predetermined protocol composed of at least a lower layer and an application layer, the method comprising the steps of:

receiving a predetermined primitive from an upper application software;

generating a communication cycle identifier (CycleID) according to the primitive;

generating a service description according to the primitive and the communication cycle identifier (CycleID);

composing an application layer protocol data unit (APDU) including the primitive; and

transmitting the APDU to the lower layer.

2. The method of claim 1, wherein the communication cycle identifier (CycleID) is generated by combining an application layer service code (ALSvcCode) and a destination address (DstAddress) included in the primitive.

3. The method of claim 2, wherein the application layer service code (ALSvcCode) comprises at least a product code and a command code.

4. The method of claim 2, wherein the communication cycle identifier CycleID is 4 bytes.

5. The method of claim 1, wherein the service description comprises an application layer service code (ALSvcCode) included in the primitive.

6. The method of claim 1, wherein the service description comprises at least an application layer service (ALService) included in the primitive, and an application layer service type (ALSvcType) according to an application layer service code (ALSvcCode).

7. The method of claim 6, wherein the application layer service type (ALSvcType) is one of a request-response-message, a request-message-only, a repeated-message and an event-message-only.

8. The method of claim 7, wherein the application layer service type (ALSvcType) comprises a first code if the application layer service (ALService) is a request-response-message, a second code for a request-message-only, a third cord for a repeated-message and a fourth code for an event-message-only.

9. The method of claim 7, wherein the first code is '0', the second code is '1', the third code is '2' and the fourth code is '3'.

10. The method of claim 1, wherein the service description comprises a communication cycle number (CycleNo) that indicates a number of communication cycles according to a kind of the primitive.

11. The method of claim 1, wherein the service description comprises the communication cycle identifier (CycleID).

12. The method of claim 1 further comprises the step of:

completing a communication cycle.

13. The method of claim 12, wherein the completion step comprises the sub-step of:

making a first decision regarding whether the network layer completing primitive (NLCompleted) is received for a service time out (SvcTimeOut) from the service execution.

14. The method of claim 13, wherein if, in the first decision step, the network layer completing primitive (NLCompleted) is received for the service time out (SvcTimeOut), the communication cycle number (CycleNo) is reduced by a predetermined value, and if not, the communication cycle number (CycleNo) is set to '0'.

15. The method of claim 11, wherein the completion step further comprises the sub-step of:

making a second decision regarding whether the communication cycle number (CycleNo) is '0'.

16. The method of claim 15, wherein if, in the second decision step, the communication cycle number (CycleNo) is '0', the service description is deleted, and if not, the communication cycle identifier (CycleID) is generated or the APDU is composed.

17. The method of claim 13, wherein the completion step further comprises the step of:

if, in the first decision step of the completion, the network layer completing primitive (NLCompleted) is received before the service time out (SvcTimeOut)

lapses or if the service time out (SvcTimeOut) lapses, deleting the service description.

18. The method according to one of claims 1 to 16, wherein the protocol is applied to a master device.

19. The method of claim 18 further comprises the step of:

after the service description is deleted, transmitting an application layer completing primitive (ALCompleted) to the application software.

20. The method of claim 18 further comprises the step of:

after the communication cycle identifier (CycleID) is generated, searching a pre-stored service description corresponding to the generated communication cycle identifier (CycleID), so as to decide, based on the search result, whether to generat the service description.

21. The method of claim 18 further comprises the steps of:

after the primitive is received, separating a communication cycle based on an application layer service code (ALSvcCode) included in the primitive; and

generating a request or a notification message according to the separated communication cycle,

wherein the communication cycle is separated and the request or the notification message is generated during the composition step of the APDU.

22. The method of claim 21, wherein the primitive is a user request primitive (UserReq).

23. The method of claim 22, wherein, a single communication cycle corresponding to the user request primitive (UserReq) is separated in the separation step, and a single request message is generated in the request message generation step.

24. The method of claim 22, wherein the communication cycle number (CycleNo) comprises '1'.

25. The method of claim 21, wherein the primitive is a user download request primitive (UserDLReq) or a user upload request primitive (UserULReq).

26. The method of claim 23, wherein a plurality of communication cycles corresponding to the primitive are separated in the separation step, and a plurality of request messages are generated in the request message generation step; and wherein a communication cycle for a next request message is processed upon completion of a communication cycle for one request message.

27. The method of claim 26, wherein the communication cycle number (CycleNo) comprises the number of partitioned data according to a download or upload procedure.

28. The method according to one of claims 1 to 12, and 17, wherein the protocol is applied to a slave device.

29. The method of claim 28 further comprises the steps of:

after the communication cycle identifier (CycleID) is generated, searching a

pre-stored service description corresponding to the generated communication cycle identifier (CycleID); and

based on the search result, standing by until the searched service description is deleted.

30. The method of claim 28 further comprises the steps of:

after the primitive is received, setting a data reception function to a disable state; and

generating an event message including an event code (EventCode) and a state variable (StateVariable) in the primitive, wherein the event message is included in the composing APDU.

31. The method of claim 28, wherein a node address of the communication cycle identifier (CycleID) comprises a node address of every master device.

32. The method of claim 30 further comprises the step of:

after the service description is deleted, setting the data reception function to an enable state.

33. The method of claim 1, wherein the protocol is a living network control protocol (LnCP).

34. A data processing method for application layer based on a protocol composed of at least a lower layer and an application layer, the method comprising the steps of:

receiving a request message receiving primitive (ReqMsgRcv) including an application layer protocol data unit (APDU) from the lower layer;

making a first decision regarding whether to process the APDU;

based on a result of the first decision, extracting a message from the APDU;

making a second decision regarding whether to generate a service description according to the request message receiving primitive (ReqMsgRcv);

based on a result of the second decision, generating the service description;

generating a user request receiving primitive (UserReqRcv) primitive including the message; and

transmitting the generated user request receiving primitive (UserReqRcv) to an application software.

35. The method of claim 34 further comprises the step of:

after the primitive receiving step, setting a data reception function to an disable state.

36. The method of claim 34, wherein the first decision is made depending on whether a value of an application layer option (ALO) field in the APDU comprises a predetermined value.

37. The method of claim 36, wherein the message is extracted using a value of an APDU header length (AHL) field in the APDU.

38. The method of claim 34, wherein the second decision is made on the

basis of a network layer service (NLService) included in the request message receiving primitive (ReqMsgRcv).

39. The method of claim 38, wherein if, in the second decision step, the network layer service (NLService) is acknowledged, the service description is generated, and if the network layer service (NLService) is non-acknowledged, the service description is not generated.

40. The method of claim 34, wherein the service description comprises an application layer service code (ALSvcCode), an application layer service type (ALSvcType), a communication cycle number (CycleNo), and a communication cycle identifier (CycleID).

41. The method of claim 40, wherein the application layer service type (ALSvcType) comprises a request-response-message.

42. The method of claim 40, wherein the application layer service code (ALSvcCode) comprises a destination address (DstAddress) included in the primitive and a command code included in the extracted message;

wherein the application layer service type (ALSvcType) comprises a network layer service (NLService) included in the primitive and a value of the application layer service code (ALSvcCode);

wherein the communication cycle number (CycleNo) comprises a fixed value by a kind of the primitive;

wherein the communication cycle identifier (CycleID) comprises the generated communication cycle identifier (CycleID).

43. The method of claim 42, wherein the application layer service type (ALSvcType) comprises a code indicating an acknowledged service.

44. The method of claim 35 or claim 39 further comprises the step of:

if a network layer service (NLService) of the request message receiving primitive (ReqMsgRcv) is non-acknowledged, setting the data reception function after the transmission of the user request receiving primitive (UserReqRcv) to an enable state.

45. The method of claim 34, wherein the protocol is a living network control protocol (LnCP).

46. A data processing method for application layer based on a protocol composed of at least a lower layer and an application layer, the method comprising the steps of:

receiving a user response sending primitive (UserResSend) from an upper application software;

generating a response message according to the user response sending primitive (UserResSend);

composing an application layer protocol data unit (APDU) including the response message; and

transmitting the APDU to the network layer using a pre-stored service description.

47. The method of claim 46 further comprises the step of:

completing a communication cycle.

48. The method of claim 47, wherein the completion step comprises the sub-steps of:

making a first decision regarding whether a service execution time from the network layer is within a service time out (SvcTimeOut); and

based on a result of the first decision, making a second decision regarding whether to receive a network layer completing primitive (NLCompleted).

49. The method of claim 48 further comprises the sub-step of:

deleting the service description.

50. The method of claim 48 or claim 49, wherein if, in the first decision step, the service execution time is for the service time out (SvcTimeOut), the second decision step is performed, and if not, the service description is deleted;

wherein if, in the second decision step, the network layer completing primitive (NLCompleted) is received, the service description is deleted, and if not, the first decision step is performed.

51. The method of claim 49 further comprises the step of:

after the service description is deleted, setting the data reception function to an enable state.

52. The method of claim 46, wherein the response message is generated using a response data (ResData) included in the user response sending primitive (UserResSend).

53. The method of claim 46, wherein the protocol is a living network control protocol (LnCP).

54. A data processing method for application layer based on a protocol composed of at least a lower layer and an application layer, the method comprising the steps of:

receiving a message receiving primitive (MsgRcv) including an application layer protocol data unit (APDU) from the lower layer;

making a first decision regarding whether to process the APDU;

based on a result of the first decision, extracting a message from the APDU;

making a second decision regarding whether a corresponding service description exists;

based on a result of the second decision, generating a predetermined primitive; and

transmitting the generated primitive to an upper application software.

55. The method of claim 54, wherein the first decision is made depending on whether a value of an application layer option (ALO) field of the APDU comprises a predetermined value.

56. The method of claim 55, wherein the message is extracted using a value of an APDU header length (AHL) field in the APDU.

57. The method of claim 54, wherein if, in the second decision step, a service description corresponding to a communication cycle identifier (CycleID)

included in the message receiving primitive (MsgRcv) is searched, the message receiving primitive (MsgRcv) comprises a response message.

58. The method of claim 57, wherein a user response primitive (UserRes) is generated in the primitive generation step.

59. The method of claim 54 further comprises the step of:

making a third decision regarding whether the extracted message is an event message,

wherein if, in the second decision step, a service description corresponding to the communication cycle identifier (CycleID) included in the message receiving primitive (MsgRcv) is not searched, the third decision step is performed.

60. The method of claim 59, wherein if, in the third decision step, the extracted message is an event message, a user event receiving primitive (UserEventRcv) is generated in the primitive generation step.

61. The method of claim 59 further comprises the step of:

if the extracted message is not an event message, ignoring the extracted message.

62. The method according to one of claims 59 to 61, wherein the third decision is made by comparing and deciding whether a command code of the extracted message is equal to a predetermined value of the event message.

63. The method of claim 54, wherein the protocol is a living network

control protocol (LnCP).